IN THE CLAIMS:

1. (Currently Amended) A programming station for generating an automation application, ~~program~~ said automation application to be executed in automation equipment and comprising an automation application program written in at least one graphic automation language, said programming station comprising:

an internal memory for storing a plurality of grammar files ~~all written in XML language~~ in text format and compliant with eXtensible Markup Language (XML) syntax,

each said grammar file comprising a description grammar describing a syntax of a respective graphic automation language, ~~written in the XML language which is recognizable in an XML application description file, said internal memory for storing at least one such XML application description file, each such XML application description file describing part of an automation program for execution in automation equipment and written in at least one graphic automation language compliant with the XML language,~~

said internal memory for storing a plurality of application description files, each said application description files being expressed in an XML language and describing part of said automation application, and

~~wherein~~ said programming station is for using at least one of said ~~description grammars~~grammar files to generate at least one of said application description files describing part of said~~such~~ automation application program,

wherein said application description files comprise an automation application program description file, an application input-output description file and an application data description file.

2.-4. (Cancelled)

5. (Currently Amended) The programming station according to claim ~~2~~1, wherein:

said grammar file comprises a Ladder language description program, wherein at least one application element of an

application is described as an object comprising at least one attribute comprising at least one of objects, parameters, variables and texts, and

the internal memory is for storing information having a tree structure.

6. (Previously Presented) The programming station according to claim 5, wherein application elements written according to the Ladder language program comprise a contact, a horizontal link, a vertical link, a coil, a short circuit, an empty cell, a function block call, a flexible function block expression, a comparison block and an arithmetical operations block.

7. (Previously Presented) The programming station according to claim 2, further comprising:

a programming station display means;

a table of rows and columns of data for displaying a graph of an object to be displayed by the programming station display means; and

a position object for defining graphic coordinates of a

position of said object to be displayed, wherein

said grammar file comprises a Sequential Function Charts language description program for describing an application in the Sequential Function Charts language, wherein at least one application element of an application is described as an object comprising at least one of a step, a transition element, a jump, a link between graphs, and a comment.

8. (Previously Presented) The programming station according to claim 2, wherein said grammar file comprises a function block language description program for describing an application in the function block description language, wherein at least one application element of an application is described as an object in the function block description language.

9. (Previously Presented) The programming station according to claim 8, wherein the at least one application element comprises any of function blocks, text boxes, links between blocks, jump instructions, labels and comments.

10. (Previously Presented) The programming station according to claim 1, comprising an Extended Markup Language handler stored in non-volatile memory, said handler for sending and receiving notifications with a management module of a decision tree model representative of an automation application expressed in the single, hierarchical and object oriented language, and also with a plurality of database managers, each manager being specific to part of an automation application stored in one of the databases.

11. (Previously Presented) An automation equipment for executing an automation program, comprising memory means for storing a plurality of automation XML application description files expressed in XML language, each said XML application description file describing at least part of an automation program and written in at least one graphic automation language compliant with the XML language, the automation equipment also comprising translation means for converting each XML application description file into a binary language that can be executed by the automation equipment.

12. (Cancelled)

13. (Previously Presented) The automation equipment according to claim 11 wherein the set of XML application description files contains an application program description file, an application input-output description file, and an application data description file.

14. (Previously Presented) The automation equipment according to claim 11, further comprising a grammar file storing a description grammar, said description grammar for translation of at least a part of such automation program from at least one graphic automation language into the XML language.

15. (Previously Presented) The automation equipment according to claim 14, further comprising means for checking that the description of an application written in the XML language satisfies a description grammar of the graphic automation language used by the automation equipment.

16. (Previously Presented) The automation equipment according to claim 14, wherein the graphic automation language used by the automation equipment includes at least one language among the Ladder language, the SFC language and the FBD language.